Development of an MM5-Based Four-Dimension Variational Analysis System for Distributed Memory Computers

Frank H. Ruggiero

Space Vehicles Directorate

Air Force Research Laboratory

Hanscom AFB, MA

Background

Objective:

Take state-of-the-art weather analysis system and make it scalable for distributed memory multi-processor machines

Goal:

To bring mesoscale Four-Dimensional Variational Analysis (4DVAR) into the realm of operational forecasting

Support

Supported by DoD High Performance Computing and Modernization Office (HPCMO)

- **Common High Performance Computing Software Support Initiative (CHSSI), project CWO-5**
- **3 year effort, started February 2000**

Participants

AFRL: Program Management

- Frank Ruggiero

AER: Code Development

- Thomas Nehrkorn
- George Modica
- Mark Cerniglia

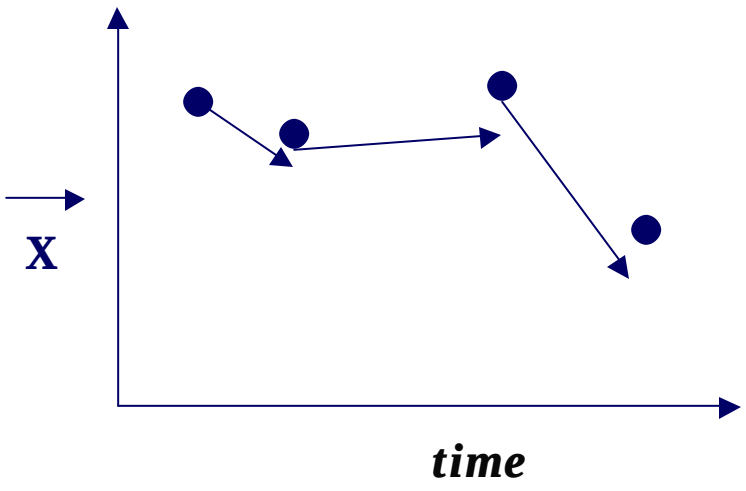
ANL: Parallel Implementation

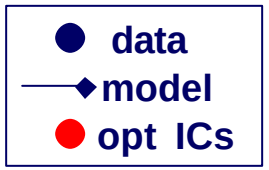
- John Michalakes

FSU: Physics Development, Test Data

- Xiaolei Zou

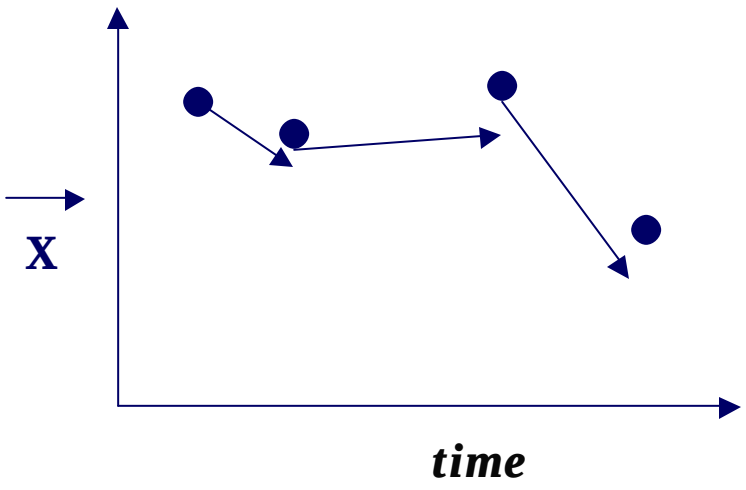
4D Data Assimilation vs 4DVAR



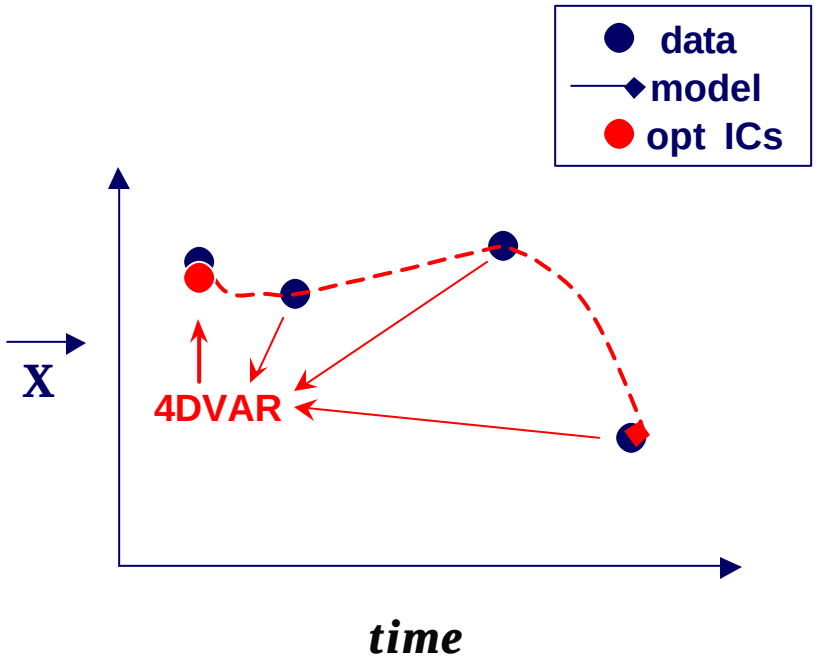


Conventional Initial Conditions

4D Data Assimilation vs 4DVAR



Conventional
Initial Conditions



Optimized Initial
Conditions

MM5 4DVAR

- *Serial Version of mesoscale 4DVAR exists*
  - *Based on Fifth Generation NCAR/PSU Mesoscale Model, Version 1 (MM5v1)*
  - *Developed by Xiaolei Zou at NCAR and released in 1997*
- *Original CWO-5 plan was to make this 4DVAR version scalable*

Migration to MM5v3

CWO-5 migrated from MM5v1 to MM5v3 (v3.3) in Oct '00

- facilitate implementation of parallel strategy
  - save development time and effort
  - increase potential speedup factor
- sync 4DVAR system with newest forecast model
  - greater community acceptance

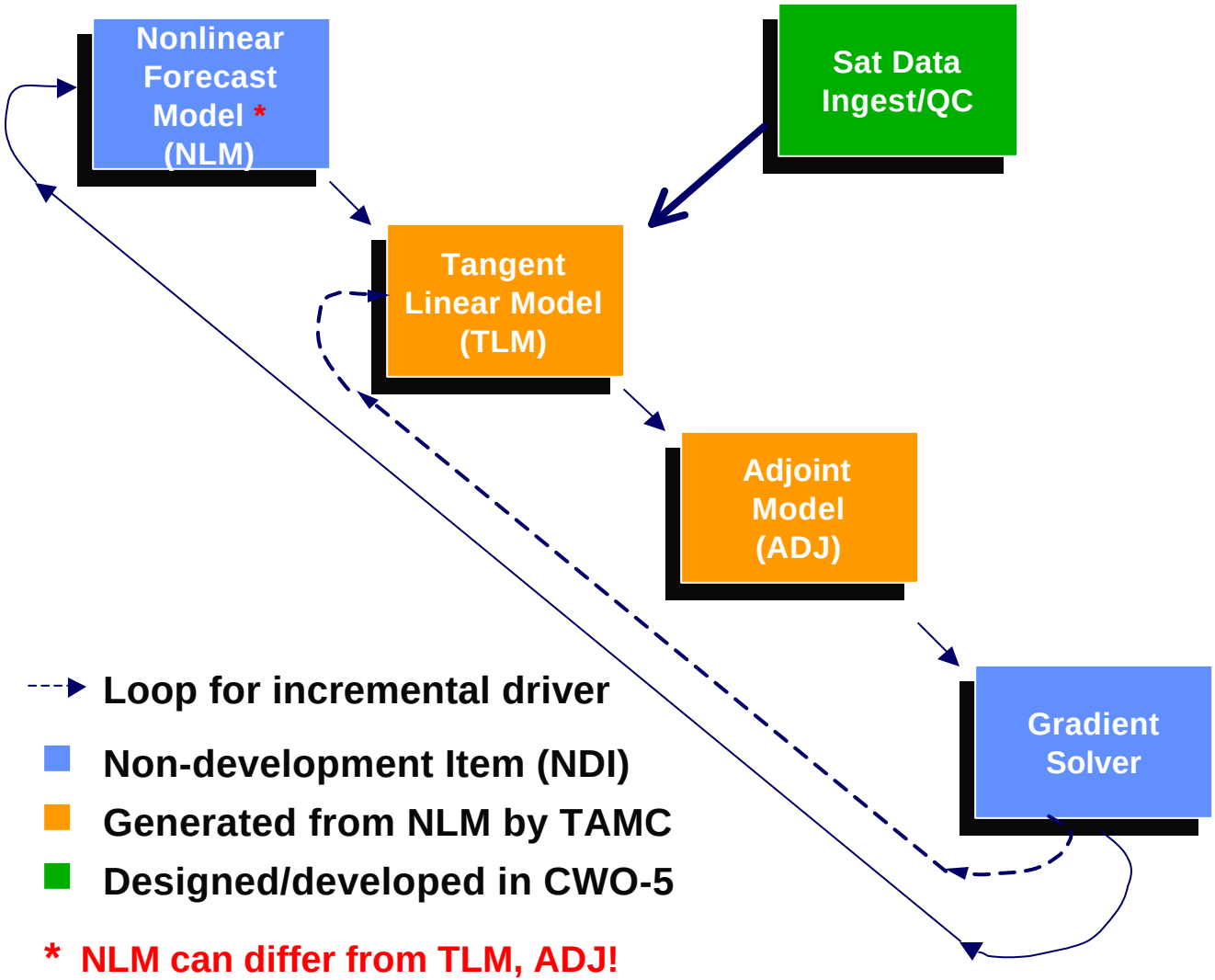
MM5 4DVAR System

Baseline code is composed of 5 CSCIs

NLM provides full physics model trajectory

TLM "predicts" state of observation vector

Adjoint model measures misfit between predicted and actual obs



The Tangent-Linear and Adjoint Model Compiler (TAMC)

TAMC is a source-to-source translator that generates Fortran code for the TLM or ADJ from the NLM code

- TAMC can be used remotely for noncommercial use, free of charge http://puddle.mit.edu/~ralf/tamc
- supports almost the full FORTRAN-77 standard and most Fortran-90 extensions to FORTRAN-77

The Tangent-Linear and Adjoint Model Compiler (TAMC)

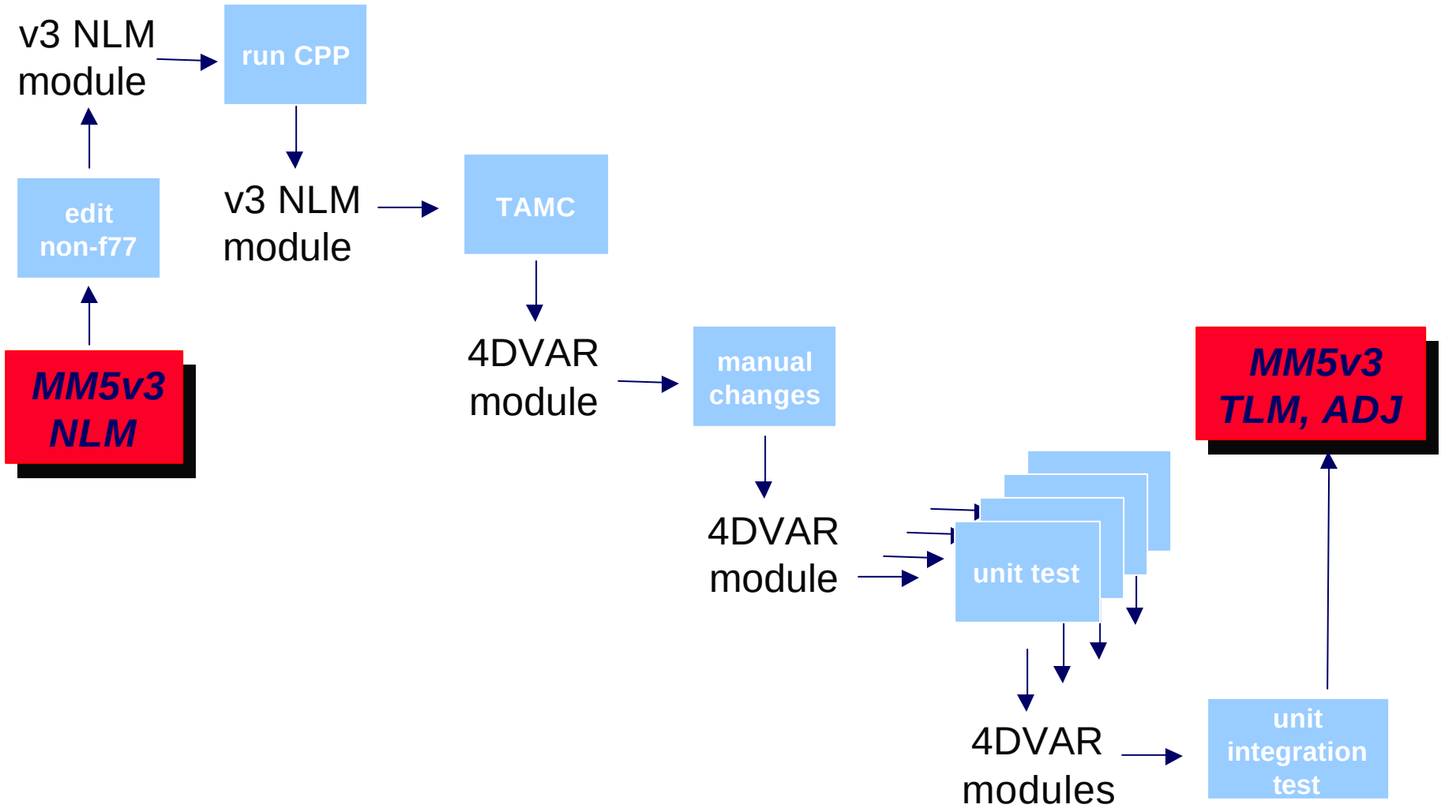
Could incorporate TAMC as part of a pre-compilation process

- would make for truly “same source” 4DVAR system
  - need maintain NLM only

We use TAMC as a development tool only

- separately maintain TLM and ADJ
  - minimizes required changes to NCAR-supported MM5 code

TLM, ADJ Code Generation and Test Procedure



Unit and Unit Integration Testing Requirements

TLM

$$\frac{\left[J(\mathbf{X}+d\mathbf{X})-J(\mathbf{X})\right]}{dJ(\mathbf{X},d\mathbf{X})}=1 \qquad \lim d\mathbf{X}\rightarrow 0$$

Where

$J(\mathbf{X})$, $J(\mathbf{X}+\delta\mathbf{X})$ are cost functions from NLM

$\delta J(\mathbf{X}, \delta\mathbf{X})$ is from TLM

ADJ

$$\left\langle \mathbf{y}, L\mathbf{x}\right\rangle = \left\langle L^{*}\mathbf{y}, \mathbf{x}\right\rangle$$

Where

$\mathbf{x}$ is TLM input

$\mathbf{y} = L(\mathbf{x})$ is TLM output

L, L^{*} are TLM, ADJ

TLM and Adjoint Development Status

Serial version of MM5v3.3 TLM completed in December 2000

Migrated TLM to MM5v3.4 in Dec '00

- **good test of our code generation/maintenance procedures**
- **v3.4 code generation and testing (TLM only) accomplished in less than one week!**

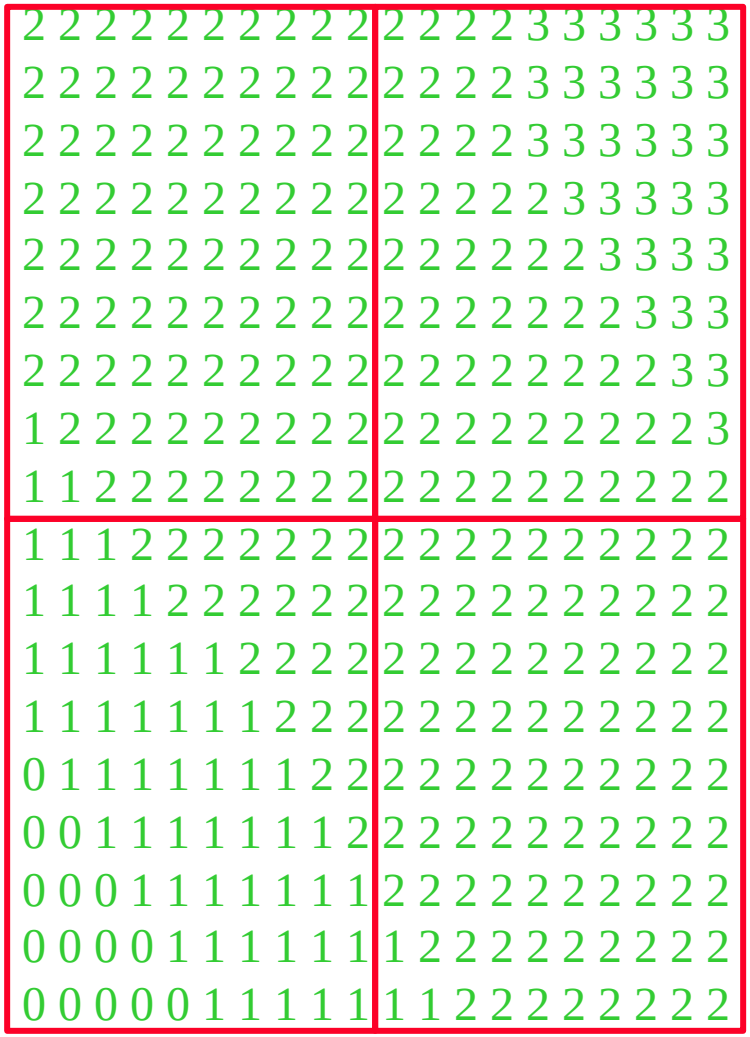
Parallel Version of TLM completed in March '01

Serial version of Adjoint model:

Coding - 100% complete

Unit testing - 90% complete

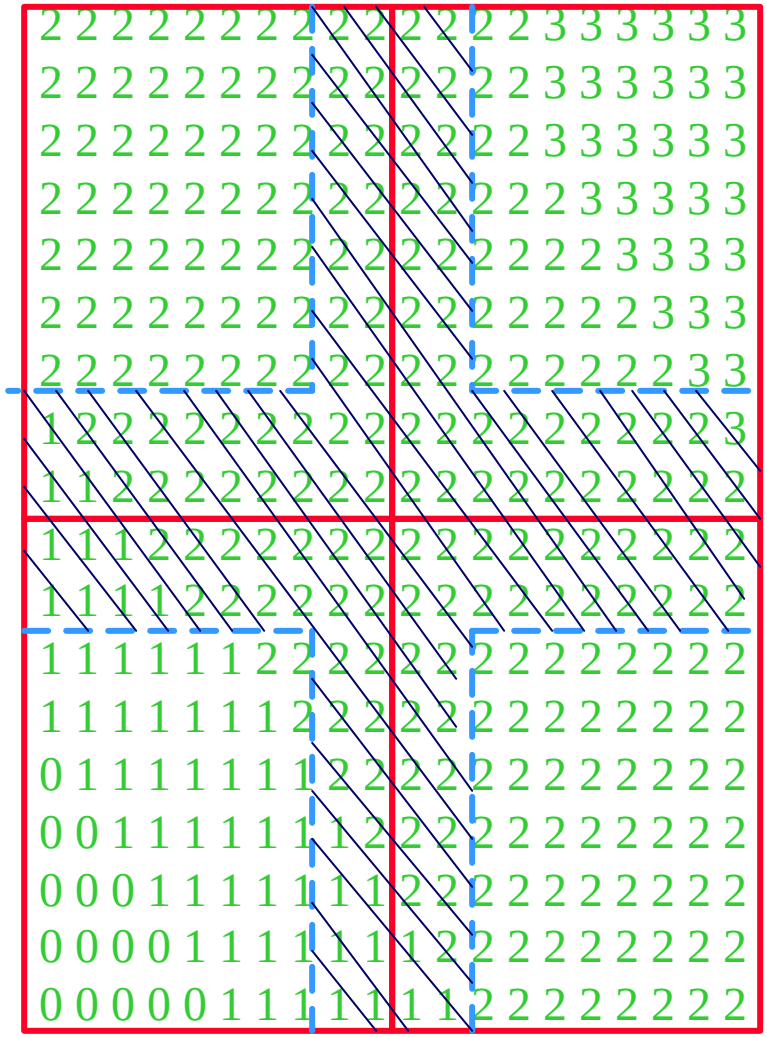
Parallelization of MM5 TLM



Follows procedure developed for MM5v3 NLM

- **2-D horizontal data decomposition**

Parallelization of MM5 TLM



Follows procedure developed for MM5v3 NLM

- **2-D horizontal data decomposition**
- **Communication of lateral boundary conditions between processors**
- **Implementation straight forward since data dependencies are similar in NLM and TLM**

Same-Source Parallel Implementation

RSL - Parallel run-time system library

- Interface with MPI, provides high-level stencil and inter-domain exchange, irregular domain decomposition, automatic local/global index translation, and distributed I/O.

FLIC - Fortran loop and index converter

- Parser-based source translation tool that automates the conversion of program loops and array indices for distributed-memory parallel computers.
- Utilizes RSL-provided loop ranges.

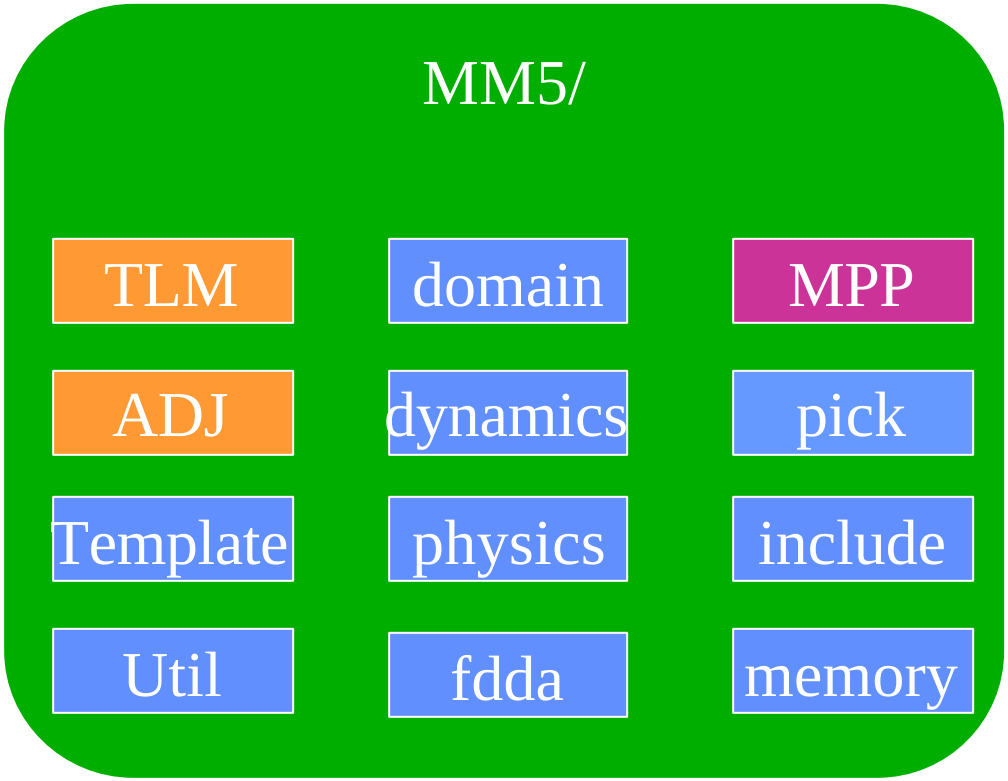
Distributed Memory (DM)-specific code fragments

DM-specific subroutines

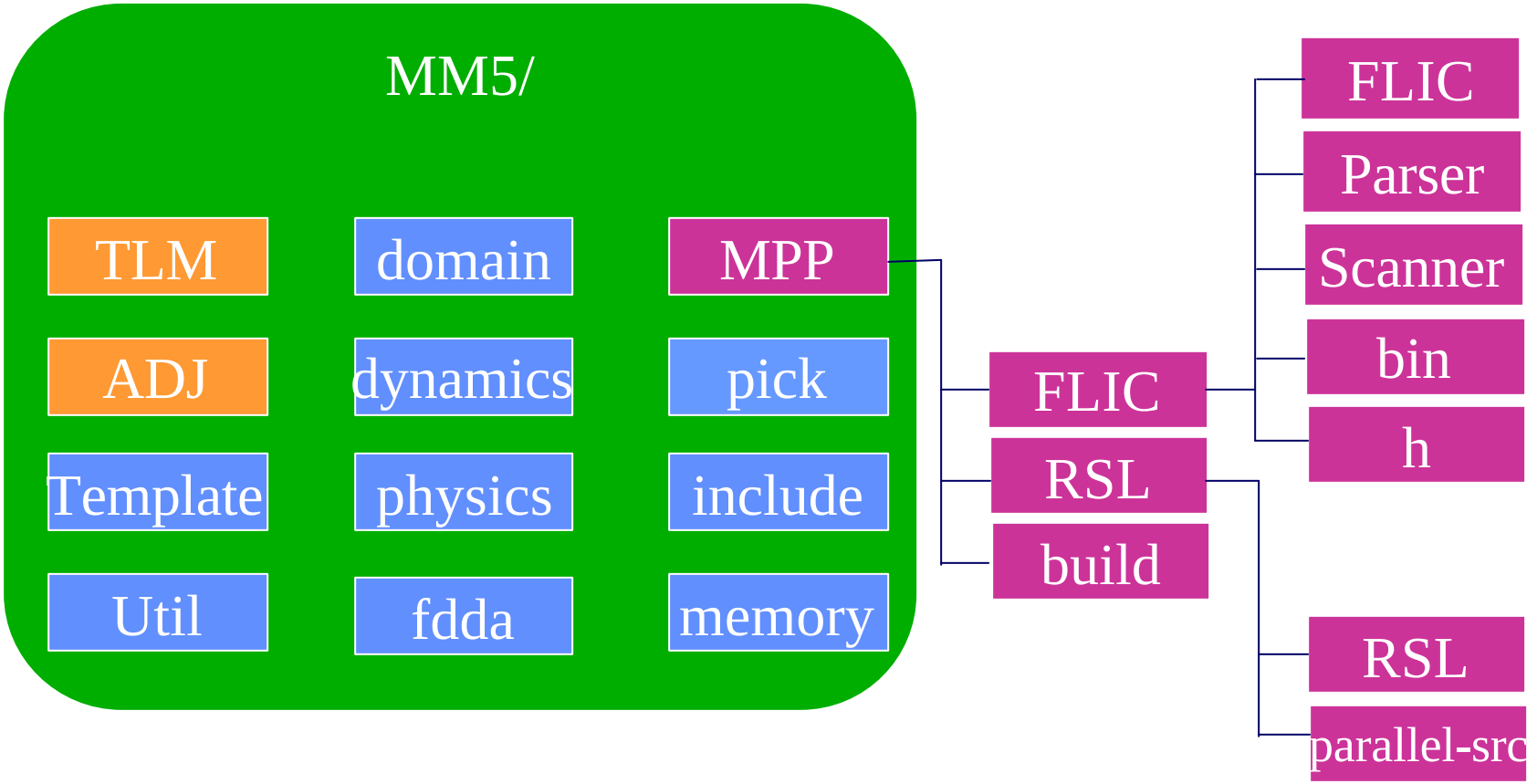
DM-related files

- Includes makefiles, tables, misc files used to install, preprocess, compile, and link code in DM-parallel mode.

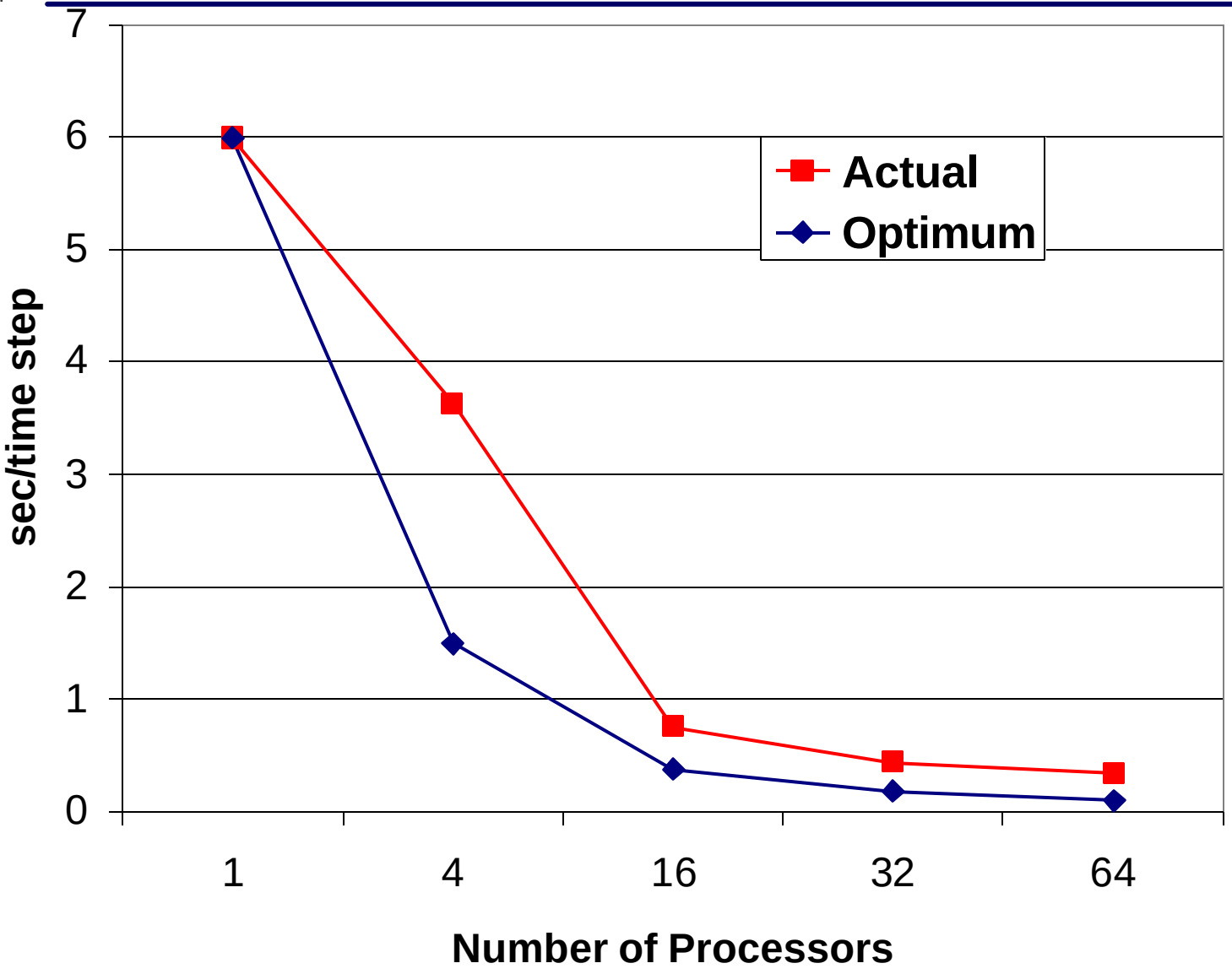
4DVAR Baseline and MPP Directory Build Structure



4DVAR Baseline and MPP Directory Build Structure



TLM Scaling Performance



TLM Multiprocessor, Cross-Platform Correctness Performance

Obtained bit-for-bit agreement (real*8) between 1-processor, 4-processor parallel, and serial runs of TLM

- **meets final project objective**

Obtained 14-digit agreement between above runs on SGI Origin, ASC IBM SP3, and DEC Alpha Workstation cluster

- **meets final project objective**

Parallelization of MM5 Adjoint

2-D horizontal data decomposition

Communication for horizontal dependencies

- Analysis of dependencies in Adjoint
  - Significantly different from NLM and TLM
- Some restructuring of data (3D tend arrays)
- Implemented with RSL and FLIC

Distributed I/O

- Model input and output (similar to NLM)
- Scratch I/O for trajectories

Remaining Tasks

Complete Adjoint model testing

Code incremental driver

Complete Adjoint model parallel code implementation

Prepare satellite observation operators

Assemble satellite data preprocessor

b-Test

- **emulate AFWA computing environment**
  - IBM SP3 computer (either at ASC or NAVO)
  - operational dataset as input (from CRDA)
- **compare forecast, timing metrics with AFWA-MM5 forecast**

Summary

CWO-5 will produce MM5 4DVAR code for DM MPP

- based on current MM5 release
- Alpha version available Fall 2001
- Beta version available Fall 2002
- Project completion Feb 2003

CWO-5 development team has put considerable effort into ensuring

- risks are minimized for successful operational implementation
- product will be well tested and of high quality
- a robust update migration path exists